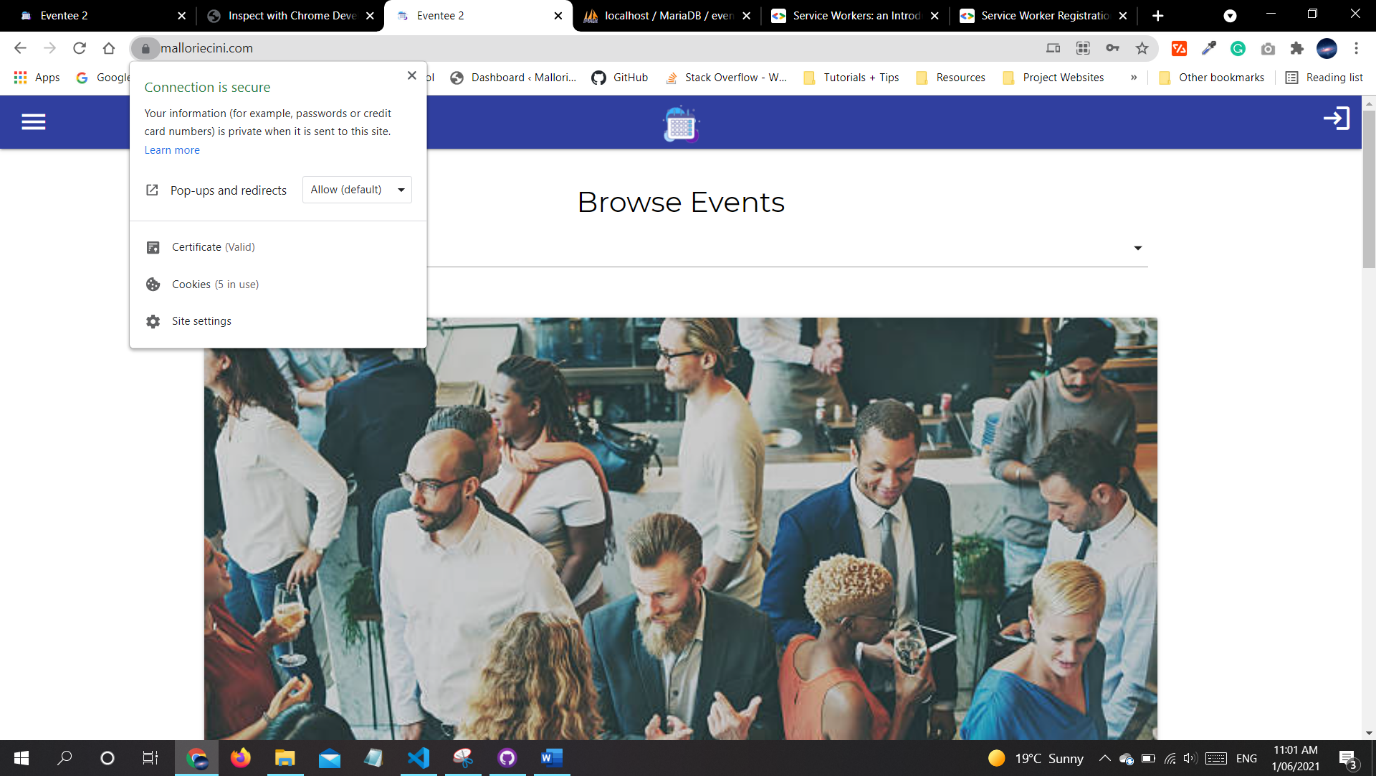
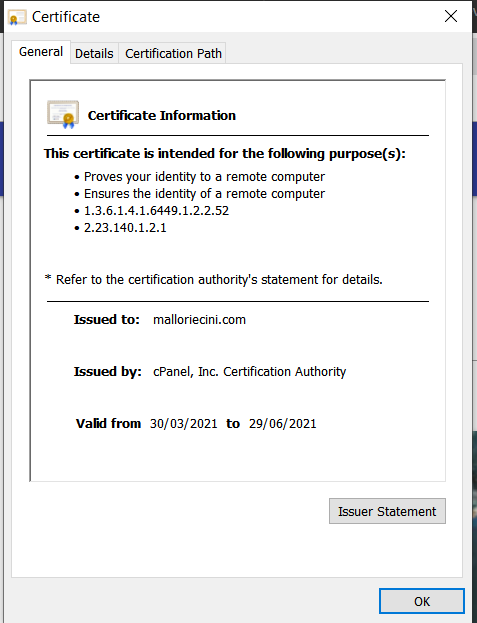
PROJ3 – Metacognition

**Part C – Security Screenshot**

**11. Test application over HTTPS connection, (screenshot evidence)**





**PART D - Metacognition**

**12. Comment on each of the 3rd party frameworks used, why was it chosen**

The frameworks chosen for this Admin Panel includes React as it is the most widely used framework and Materialize a layout framework that was used for the front end of my web application, I also used React Hooks forms in order to validate my form values and incorporating the standard html validation.

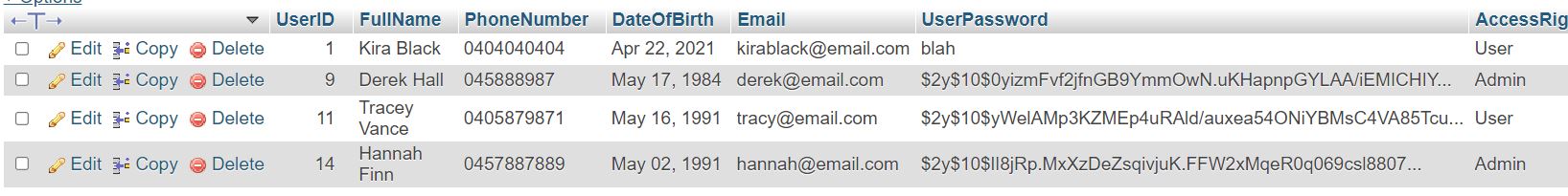
**13. What other technologies did you investigate in order to settle on a path**

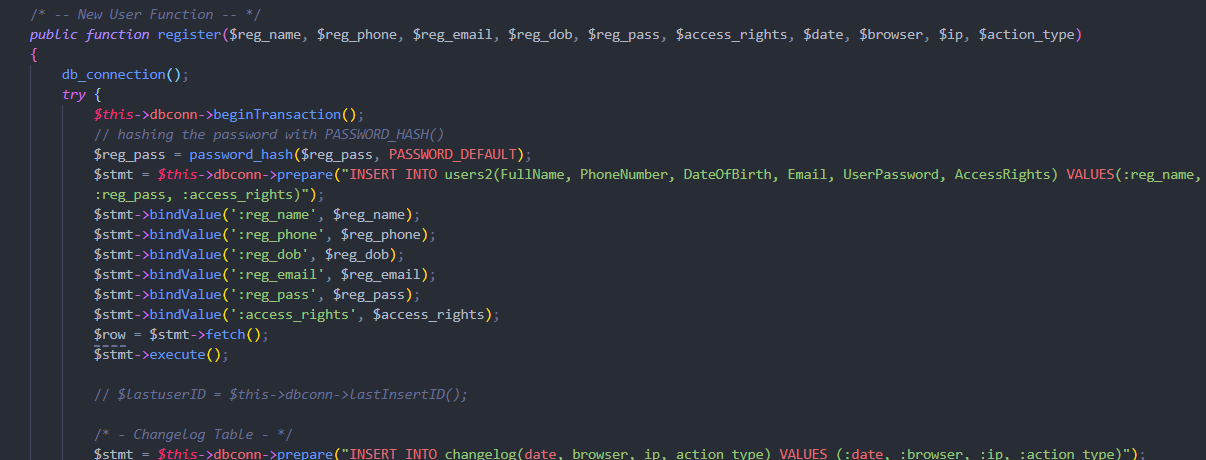
Originally, I looked at Mapbox while deciding on which maps api would best fit the purpose of my application. Another technology I investigated was Angular, I was deciding between that and React for the backend side of the admin panel. I ultimately decided on React as it was the framework most sought after in the job industry.

**14. Describe the rules by which your authentication restricts access. Comment in code.**

It restricts access to certain elements by changing the display of certain links that logged in users should only have access to by using both Sessions and local storage for my normal application and I used React state in order to retain logged in status along with sessions + local storage for my admin panel. As for ip whitelisting I have created an if statement within my ws.php file to check if the user’s domain url matches the parameters within the If statement and if they don’t then the request dies and they are unable to access it. I have also got an if statement that checks the access rights of the user when they try to login to the admin panel located within the adminLogin function.

Authentication with passwords:





**15. Describe why you chose this particular encryption technology**

I chose to rely on both the materialize validation along with incorporating react hook forms into the application as it also uses the default HTML validation techniques as well as aligning with existing HTML standards for validating forms. I also added another layer of security with JavaScript and PHP with password\_hash along with password\_verify to check that the value is equal to the value in the database and also if statements to check if empty.